

Nama : RIZQILLAH
NIM : 1957301020
MK : Prak. Pemrograman Lanjut

RETURN VALUE

1.1 Dasar Teori Return Value

Return value adalah metode yang akan mengembalikan nilai yang dikembalikan, kemudian nilai tersebut akan diterima oleh si pemanggil fungsi. Function(method) dalam pemrograman memiliki 2 jenis, yaitu function yang mengembalikan nilai, dan function tanpa pengembalian nilai. Dan return value biasanya digunakan pada function yang mengembalikan nilai.

Statement return digunakan untuk keluar dari sebuah fungsi (method). Statement return memiliki dua bentuk : menggunakan sebuah nilai, dan tidak memberikan nilai. Untuk memberikan sebuah nilai, cukup berikan nilai (atau ekspresi yang menghasilkan sebuah nilai) sesudah return. Contohnya,

```
return ++count;
```

atau

```
return "Hello";
```

Tipe data dari nilai yang diberikan harus sama dengan tipe dari fungsi yang dideklarasikan. Ketika sebuah method void dideklarasikan, gunakan bentuk return yang tidak memberikan nilai. Contohnya,

```
return;
```

Adapun contoh function yang memiliki nilai kembalian pada pemrograman Java adalah suatu function yang memiliki tipe data, contohnya :

```
public int Nilai(int nilai) {  
    return nilai;  
}
```

Pada contoh program diatas, dapat kita lihat itu adalah sebuah function(method) yang akan mengembalikan nilai pada variabel nilai ketempat si pemanggil function tersebut. Dan dapat juga kita lihat, bahwa disaat kita membuat function yang akan mengembalikan nilai, maka kita harus mendeklarasikan function tersebut dengan suatu tipe data primitif. Dan public pada function tersebut menandakan hak akses nya bisa diakses oleh seluruh class lainnya.

Adapun contoh function yang tidak mengembalikan nilai kepada si pemanggilnya adalah sebagai berikut :

```
public void Contoh() {  
    /*....  
    Statement berada  
    */  
}
```

Sebuah function yang tidak mengembalikan nilai pada Java pasti akan menggunakan tipe void.

1.2 Contoh Program

Coding :

```
import java.util.Scanner;  
public class ReturnValue {  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  
        System.out.print("Input Nilai 1 : ");  
        int a = input.nextInt();  
        System.out.print("Input Nilai 2 : ");  
        int b = input.nextInt();  
        System.out.println(Tambah(a, b));  
    }  
  
    public static int Tambah(int a, int b) {  
        return a + b;  
    }  
}
```

Analisis :

Program diatas adalah suatu program yang menggunakan konsep return value pada sebuah fungsi yang akan mengembalikan hasil pertambahan ketempat si pemanggil fungsi tersebut.

```
import java.util.Scanner;
```

➔ Mengimport Scanner agar bisa melakukan inputan dari keyboard.

```
public class ReturnValue {  
    public static void main(String[] args) {
```

➔ Membuat class public dengan nama ReturnValue. Dan membuat fungsi main yang berarti program membaca dari ini awalnya.

```
Scanner input = new Scanner(System.in);
```

- Membuat objek dari Scanner yang nantinya digunakan untuk melakukan proses input dari keyboard oleh user.

```
System.out.print("Input Nilai 1 : ");  
int a = input.nextInt();
```

- Meminta user untuk menginputkan nilai pertama yang nantinya disimpan pada variabel a yang bertipe data integer, oleh karenanya, inputan yang diperbolehkan hanya integer, makanya digunakan nextInt().

```
System.out.print("Input Nilai 2 : ");  
int b = input.nextInt();
```

- Meminta user menginput nilai kedua yang akan disimpan pada variabel b.

```
System.out.println(Tambah(a, b));
```

- Mencetak hasil yang diperoleh ketika memanggil fungsi Tambah yang diisi parameter dengan nilai a dan b.

```
}
```

```
public static int Tambah(int a, int b) {
```

- Membuat function dengan hak akses public, dan bertipe static dikarenakan agar bisa dijalankan di main yang juga bertipe static, maka fungsi yang akan dipanggil juga harus bersifat static. Fungsi tersebut bertipe data int dengan nama Tambah dan menerima 2 parameter, yaitu int a, dan int b.

```
return a + b;
```

- Mengembalikan nilai pertambahan variabel a + variabel b.

```
}
```

```
}
```

1.3 Return yang Berfungsi sebagai Fungsi Rekursif

Adapun dalam pembuatan suatu program yang berbentuk rekursif, program yang dibuat akan melakukan suatu pemanggilan terhadap method, atau dirinya sendiri. Dan adapun contoh program faktorial yang berbentuk rekursif sebagai berikut :

```

public class FaktorialRecur {
    static int factorial(int n) {
        if (n == 1) {
            return 1;
        } else {
            return factorial(n - 1) * n;
        }
    }

    public static void main(String args[]) {
        int n = 5;
        System.out.println(factorial(n));
    }
}

```

Hasil :

```

120
BUILD SUCCESSFUL (total time: 0 seconds)

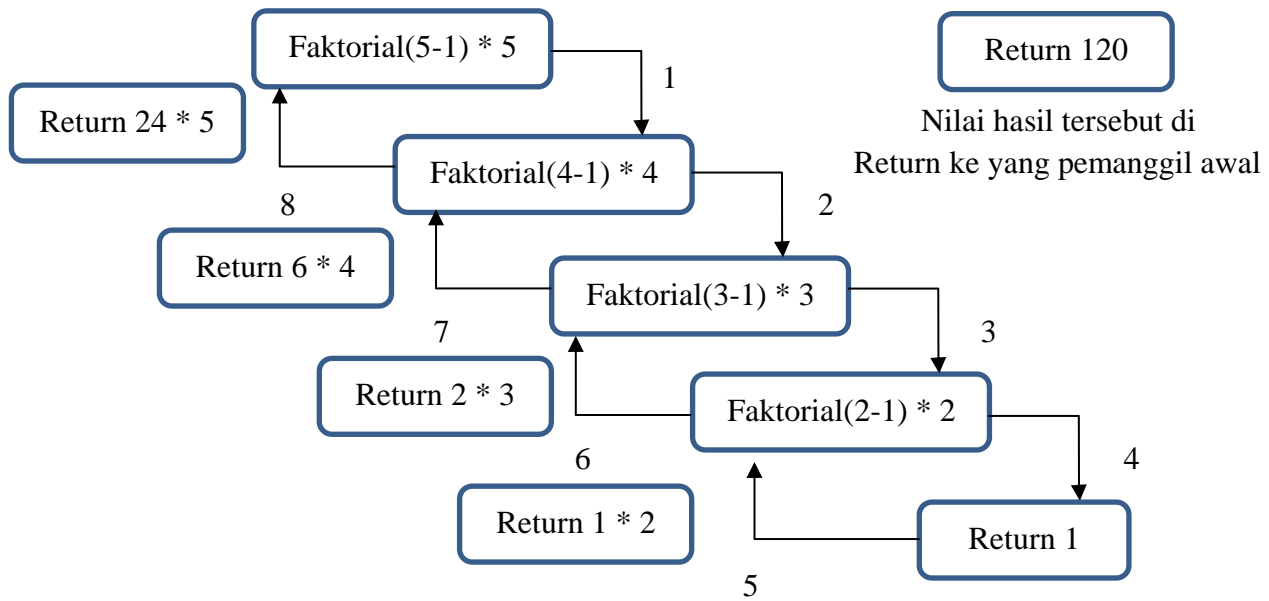
```

Analisis :

Adapun penjelasan dari metode program diatas adalah :

- ➔ Menginisialisasi variabel n bertipe data integer dengan value = 5.
- ➔ Menampilkan hasil dari pemanggilan function factorial yang diisi parameter nilai variabel n.
- ➔ Function factorial menerima nilai parameter kemudian menyimpan pada variabel n.
- ➔ Mengecek apakah $n == 1$, jika true maka akan langsung return 1, akan tetapi nilai $n = 5$, hasilnya false.
- ➔ Dan di else, program melakukan pemanggilan method faktorial yang diisi dengan nilai $5 - 1 = 4$. Kemudian nilai tersebut diterima oleh parameter n.
- ➔ Dan mengecek apakah $4 == 1$, hasil false, maka memanggil function factorial dengan nilai parameter $4 - 1 = 3$. Dan diterima oleh variabel n.
- ➔ Mengecek apakah $3 == 1$, false, maka melakukan pemanggilan function dengan nilai $3 - 1 = 2$. Dan diterima oleh variabel n.
- ➔ Mengecek apakah $2 == 1$, false, maka melakukan pemanggilan function dengan nilai $2 - 1 = 1$. Dan diterima oleh variabel n.
- ➔ Mengecek apakah $1 == 1$, hasilnya true, oleh karenanya mengembalikan nilai 1 pada si pemanggil sebelumnya.

Adapun penggambaran operasi $\text{return factorial}(n-1) * n$ sebagai berikut :



KESIMPULAN

Dapat disimpulkan bahwa return value adalah sebuah metode untuk keluar dari fungsi yang sedang berjalan, atau biasa digunakan untuk mengembalikan nilai yang telah dieksekusi didalam fungsi tersebut, kemudian mengembalikan nilai ke yang pemanggil fungsi tersebut.

Dalam pembuatan fungsi rekursif pada suatu program. Yang akan digunakan agar program tersebut bersifat rekursif adalah metode return value, akan tetapi sebelum melakukan return value pada yang pertama memanggil fungsi tersebut, return yang dilakukan adalah memanggil fungsinya sendiri dan akan mencari nilai sampai mendapati return nilai, contohnya seperti program sebelumnya yaitu jika return tersebut terus-menerus memanggil fungsinya, dan diakhir akan mendapati nilai return 1, dan nilai return 1 tersebut akan dikalikan dengan operasi perkalian di return, dan mengembalikan nilai perkalian tersebut sampai ke si pemanggilnya dan seterusnya sampai hasil didapatkan.

Metode rekursif ini mirip seperti metode stack, yaitu FILO(First In Last Out). Dikarenakan nilai yang duluan dimasukkan akan mendapati hasilnya yang paling terakhir, akibat fungsi rekursif diawal berjalan, akan melakukan operasi terus-menerus sampai menemukan return nilai yang pasti. Dan setelah menemukan nilai yang pasti, return akan kembali dan kembali lagi ketempat si pemanggil asalnya hingga program selesai.